Vector Reachability Problem in $\mathrm{SL}(2,\mathbb{Z})$ *

Igor Potapov[1] and Pavel Semukhin[2]

1 **Department of Computer Science, University of Liverpool**
Liverpool, United Kingdom
potapov@liverpool.ac.uk

2 **Department of Computer Science, University of Liverpool**
Liverpool, United Kingdom
semukhin@liverpool.ac.uk

Abstract

This paper is showing the solution for two open problems about decidability of vector reachability problem in a finitely generated semigroup of matrices from $\mathrm{SL}(2,\mathbb{Z})$ and the point to point reachability (over rational numbers) for fractional linear transformations, where associated matrices are form $\mathrm{SL}(2,\mathbb{Z})$. The approach of solving reachability problems is based on analysis of reachability paths between points following the translation of numerical reachability problems into computational and combinatorial problems on words and formal languages.

1998 ACM Subject Classification F.2.1 Numerical Algorithms and Problems

Keywords and phrases Decidability, Matrix Semigroup, Vector Reachability Problem, Special Linear Group, Linear Fractional Transformation

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

The decision problems on matrices were intensively studied form 1947 when A.Markov showed the connection between classical computations and problems for matrix semigroups [19]. Moreover matrix products are playing essential role in representation of various computational processes, i.e. linear recurrent sequences [14, 21, 22], arithmetic circuit [11], hybrid and dynamical systems [20, 2], probabilistic and quantum automata [6], stochastic games, broadcast protocols [10], optical systems, etc. New algorithms for solving reachability problems in matrix semigroups can be incorporated into software verification tools and used for analysis of mathematical models in physics, chemistry, biology, ecology, and economics.

Unfortunately, many computational problems for matrix semigroups are inherently difficult to solve even when the problem are considered in dimension two, and most of these problems become undecidable in general starting from dimension three or four. The examples of such problems are the membership problem (including the special cases of the Mortality and Identity problems), vector reachability, scalar reachability, freeness problems and emptiness of matrix semigroups intersection [5]. All above problems are tightly connected including two central problems:

- **The membership problem.** Let S be a given finitely generated semigroup of $n \times n$ matrices. Determine whether a matrix M belongs to S. In other words, determine whether there exists a sequence of matrices $M_1, M_2, \ldots, M_k$ in S such that $M_1 \cdot M_2 \cdot \ldots \cdot M_k = M$

* This work was partially supported by EPSRC grant "Reachability problems for words, matrices and maps" (EP/M00077X/1).

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–15

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- **The vector reachability problem.** Let $\mathbf{x}$ and $\mathbf{y}$ be two vectors and S be a given finitely generated semigroup of $n \times n$ matrices. Decide whether there is a matrix $M \in S$ such that $M \cdot \mathbf{x} = \mathbf{y}$.

The vector reachability problem can be seen as a parameterized version of the membership problem, where some elements of M are either independent variables or variables linked by some equations. In contrast to the original membership problem, where all values of a matrix M are defined as constants, in vector reachability we may have an infinite set of matrices that can transform a vector $\mathbf{x}$ to $\mathbf{y}$. Thus the decidability results for the membership could not be directly applied to the vector reachability problem.

Most of the problems such as membership, vector reachability and freeness are undecidable for 3×3 integer matrices. The undecidability proofs in matrix semigroups are mainly based on various techniques and methods for embedding universal computation into three and four dimensional matrices and matrix products. The case of dimension two is the most intriguing since there is some evidence that if these problems are undecidable, then it cannot be proved using a construction similar to the one used for dimension 3 and 4. In particular there is no injective semigroup morphism from pairs of words over any finite alphabet (with at least two elements) into complex 2×2 matrices [7], which means that the coding of independent pairs of words in 2×2 complex matrices is impossible and the exact encoding of the Post Correspondence Problem or a computation of the Turing Machine cannot be used directly for proving undecidability in 2×2 matrix semigroups over $\mathbb{Z}$, $\mathbb{Q}$ or $\mathbb{C}$. The only undecidability for the vector reachability and the membership problems has been shown in the case of 2×2 matrices over hypercomplex numbers (quaternions) [3].

The main hypothesis is that problems for 2×2 matrix semigroups over integers, rationals or complex numbers could be decidable, but it is still very little known about the status of these problems. Recently there was some progress on the *Membership problem*, which was shown to be decidable in case of $\mathrm{SL}(2,\mathbb{Z})$ and Mortality in $\mathbb{Z}^{2\times 2}$ [9]. Later the decidability of the *Freeness problem* (i.e. decide whether each element can always be expressed in terms of a unique product) was shown for $\mathrm{SL}(2,\mathbb{Z})$ [8]. On the other hand the Mortality, Identity and vector reachability were shown to be at least NP-hard for $\mathrm{SL}(2,\mathbb{Z})$ in [5, 4], but in the modular group case the membership is shown to be decidable in polynomial time by Gurevich and Schupp [12].

This paper is showing the solution for two open problems about the decidability of the vector reachability problem in a finitely generated semigroup of matrices from $\mathrm{SL}(2,\mathbb{Z})$ and the point to point reachability (over rational numbers) for fractional linear transformations $f_M(x) = \frac{ax+b}{cx+d}$, where associated matrix $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathrm{SL}(2,\mathbb{Z})$. The approach of solving the reachability problems in 2×2 matrix semigroups is based on the analysis of reachability paths between vectors or points following the translation of numerical reachability problems into computational and combinatorial problems on words and formal languages.

The decidability proof for vector reachability problem in dimension two presented in this paper is the first nontrivial new result for solving vector reachability problem since 1996 [1] when it was shown that the problem is decidable for any commutative matrix semigroup in any dimension. In case of non-commutative matrices the problem is known to be undecidable already for integer matrices in dimension three [13] and decidable for block monomial matrices over elements from a commutative semigroup [16], which can be seen as an extension of [1]. The paper is organized as follows. In the second section we give main definitions and provide intuitive explanation about the decidability results presented in this paper. After that in the third section we give a full formal proof leaving a few technical lemmas in the appendix.

2 Preliminaries

▶ **Definition 1.** With each matrix $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathrm{SL}(2,\mathbb{Z})$ we associate a fractional linear map (also called Möbius transformation) $f_M : \mathbb{Q} \to \mathbb{Q}$ defined as $f_M(x) = \frac{ax+b}{cx+d}$.

Note that we have $f_{M_1} \circ f_{M_2} = f_{M_1 M_2}$ for any matrices M_1 and M_2.

Let $M_1, \ldots, M_n$ be a finite collection of matrices. Then $\langle M_1, \ldots, M_n \rangle$ denotes the multiplicative semigroup generated my $M_1, \ldots, M_n$.

▶ **Definition 2.** The *vector reachability problem* in $\mathrm{SL}(2,\mathbb{Z})$ is defined as follows: Given two vectors $\mathbf{x}$ and $\mathbf{y}$ with integer coefficients and a finite collection of matrices $M_1, \ldots, M_n$ from $\mathrm{SL}(2,\mathbb{Z})$, decide whether there exists a matrix $M \in \langle M_1, \ldots, M_n \rangle$ such that $M\mathbf{x} = \mathbf{y}$.

▶ **Definition 3.** The *reachability problem by fractional linear transformations* in $\mathrm{SL}(2,\mathbb{Z})$ is defined as follows: Given two rational numbers x and y and a finite collection of matrices $M_1, \ldots, M_n$ from $\mathrm{SL}(2,\mathbb{Z})$, decide whether there exists a matrix $M \in \langle M_1, \ldots, M_n \rangle$ such that $f_M(x) = y$.

The mains result of our paper is that the vector reachability problem and the reachability problem by fractional linear transformations for $\mathrm{SL}(2,\mathbb{Z})$ are decidable (Theorems 14 and 15). Both proofs use the same pattern. First, note that any matrix M from $\mathrm{SL}(2,\mathbb{Z})$ can be expressed as product of matrices $S = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ and $R = \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}$ [18]. Thus we identify any $M \in \mathrm{SL}(2,\mathbb{Z})$ with a corresponding word w in the alphabet $\Sigma = \{S, R\}$.

The main idea of both proofs is to show that the set of matrices that satisfies the equation $M\mathbf{x} = \mathbf{y}$ or $f_M(x) = y$ corresponds to a regular language (Theorems 9 and 11). On the other hand, the language that corresponds to the semigroup $\langle M_1, \ldots, M_n \rangle$ is also regular. Indeed, if M_i corresponds to the word w_i, then $\langle M_1, \ldots, M_n \rangle$ corresponds to the language $(w_1 + \cdots + w_n)^*$. The last step of the proof is to show that the emptiness problem of the intersection of two such languages is decidable (Proposition 13).

Here is a more detailed description of our proofs. Let $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$. To show that the equation $M\mathbf{x} = \mathbf{y}$ defines a regular language we must solve the following system of equations:

$$x_1 a + x_2 b = y_1 \qquad x_1 c + x_2 d = y_2 \qquad ad - bc = 1$$

It contains four unknown variables a, b, c, d and three equations, two of which are linear and one is non-linear. We can solve this system explicitly. Say if we choose b as a free parameter, then we can write the solution as

$$a = \frac{y_1 - x_2 b}{x_1}, \quad d = \frac{x_1 + y_2 b}{y_1}, \quad c = \frac{y_1 y_2 - x_1 x_2 - x_2 y_2 b}{x_1 y_1}.$$

Since we are interested only in integer solutions, we need to find the values of b for which the above expressions are equal to integer numbers. Such values of b must satisfy the following congruence equations:

$$\begin{aligned} x_2 b &\equiv y_1 \mod x_1 \\ y_2 b &\equiv -x_1 \mod y_1 \\ x_2 y_2 b &\equiv y_1 y_2 - x_1 x_2 \mod x_1 y_1 \end{aligned}$$

By Lemma 5 the above system either has no solutions or it has a solution of the form $b \equiv b_2 \mod b_1$, where $b_1 \mid x_1 y_1$. That is $b = b_1 t + b_2$, where $t \in \mathbb{Z}$. Thus all coefficients of the matrix M are linear functions of t. In Proposition 8 we will show that such matrices can be written in the form $M = BT^{kt}C$, where B, C and T are some matrices from $\mathrm{SL}(2,\mathbb{Z})$, k is a fixed integer number and $t \in \mathbb{Z}$ is a free parameter. Now it is not hard to see that such equation defines a regular language.

We will use a similar approach to show that the equation $f_M(x) = y$ also defines a regular language. If we let $x = \frac{m_0}{n_0}$ and $y = \frac{m_1}{n_1}$, then we need to solve the following system of equations

$$\frac{a\frac{m_0}{n_0} + b}{c\frac{m_0}{n_0} + d} = \frac{m_1}{n_1} \quad \text{and} \quad ad - bc = 1.$$

This time we have only two equations and four unknowns. If we choose c and d as free parameters, then we can write its solution as

$$a = \frac{cm_1}{n_1} + \frac{n_0}{cm_0 + dn_0} \quad \text{and} \quad b = \frac{dm_1}{n_1} - \frac{m_0}{cm_0 + dn_0}.$$

To find the values of c and d for which the above expressions are equal to integer numbers, we use the following trick. We will show that there are only finitely many possible values of the denominator $cm_0 + dn_0$ for which a and b can be equal to integer numbers. Thus c becomes a linear function of d, and hence all values a, b, c, d become linear functions of one parameter. Therefore, by Proposition 8, the solution of $f_M(x) = y$ can be written as a finite union of the sets $\{C_i T^{s_i t} D_i \ : \ t \in \mathbb{Z}\}$, for $i = 1, \ldots, n$. Here T, C_i and D_i are matrices form $\mathrm{SL}(2,\mathbb{Z})$ and s_i are fixed integer numbers. So the solution of the equation $f_M(x) = y$ indeed defines a regular language.

The final step is to show that there is an algorithm that decides whether the intersection of two regular subsets of $\mathrm{SL}(2,\mathbb{Z})$ is empty or not. Our idea relies on the fact that the intersection of two regular languages is regular, and that the emptiness problem for regular languages in decidable. The problem here is that we cannot apply these facts directly because for each matrix $M \in \mathrm{SL}(2,\mathbb{Z})$ there are infinitely many words $w \in \{S, R\}^*$ that correspond to M, and only some of them may appear in the given language. However there is only one *reduced* word that corresponds to M, that is, the word that does not have a substring of the form SS or RRR. So our solution is to take any automaton A and turn it into a new automaton $\widetilde{A}$ that accepts the same language as A plus all reduced words w that correspond to non-reduced words w' accepted by A.

Note that in $\mathrm{SL}(2,\mathbb{Z})$ we have $S^2 = R^3 = -I$. Thus to construct $\widetilde{A}$ we add to A a new ε-transition from a state q_1 to a state q_2 if there is a run of A from q_1 to q_2 labelled by SS or RRR. We will apply this procedure iteratively until no new ε-transitions can be added. However we need to keep track of sign changes when we add new ε-transitions. To achieve this we will use *signed automata*, which are slight modifications of the usual finite automata but they take into account such sign changes.

Now to solve the emptiness problem for the intersection of two regular languages L_1 and L_2, we take the signed automata A_1 and A_2 that accept L_1 and L_2, respectively, and construct new automata $\widetilde{A}_1$ and $\widetilde{A}_2$ as described above. Finally we check whether $L(\widetilde{A}_1) \cap L(\widetilde{A}_2) \neq \emptyset$.

3 Main results

We will need the following two lemmas whose proofs can be found in the Appendix.

▶ **Lemma 4.** *Consider a linear congruence equation* $ax \equiv b \mod n$. *If* $\gcd(a,n) \nmid b$, *then the equation has no solution. If* $\gcd(a,n) \mid b$, *then all solutions of the equation can be written in the form* $x \equiv c \mod \frac{n}{\gcd(a,n)}$ *for some* c. *Moreover, there is a polynomial time algorithm that determines whether such equation has a solution and if so, finds it.*

▶ **Lemma 5.** *Consider a system of two linear congruence equations*

$$\begin{aligned} a_1 x &\equiv b_1 \mod n_1 \\ a_2 x &\equiv b_2 \mod n_2 \end{aligned} \tag{1}$$

Such system either has no solution, or all its solutions are of the form $x \equiv c \mod n$ *for some* c *and* $n \mid n_1 n_2$. *Moreover, there is a polynomial time algorithm that determines whether (1) has a solution and if so, finds it.*

▶ **Proposition 6.** *Let* $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ *and* $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ *be vectors from* $\mathbb{Z} \times \mathbb{Z}$ *and consider the matrix equation* $M\mathbf{x} = \mathbf{y}$, *where* M *is an unknown matrix from* $\mathrm{SL}(2,\mathbb{Z})$. *Then either this equation does not have a solution or all its solutions are given by* $M = tA_1 + A_2$, *where* t *is any integer number and* A_1, A_2 *are some matrices from* $\mathbb{Z}^{2\times 2}$. *Moreover, there is a polynomial time algorithm that determines whether such equation has a solution and if so, finds it.*

Proof. Let $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and suppose that M satisfies the equations $M\mathbf{x} = \mathbf{y}$ and $\det(M) = 1$. So we have the following system of equations:

$$x_1 a + x_2 b = y_1 \tag{2}$$
$$x_1 c + x_2 d = y_2 \tag{3}$$
$$ad - bc = 1 \tag{4}$$

Assume that $\mathbf{x} \neq \mathbf{0}$ since otherwise the proposition is obvious. Without loss of generality, suppose that $x_1 \neq 0$. In this case we have

$$a = \frac{y_1 - x_2 b}{x_1}, \quad c = \frac{y_2 - x_2 d}{x_1}.$$

Substituting these values for a and c in (4), we obtain

$$(y_1 - x_2 b)d - (y_2 - x_2 d)b = x_1$$

or $y_1 d - y_2 b = x_1$. If $y_1 = y_2 = 0$, then there is no solution because by assumption $x_1 \neq 0$. Again, without loss of generality, assume that $y_1 \neq 0$. Hence $d = \dfrac{x_1 + y_2 b}{y_1}$. If we choose b as a free parameter, then the general solution of the system of equations (2)–(4) will be:

$$a = \frac{y_1 - x_2 b}{x_1}, \quad d = \frac{x_1 + y_2 b}{y_1},$$
$$c = \frac{y_2 - x_2 \frac{x_1 + y_2 b}{y_1}}{x_1} = \frac{y_1 y_2 - x_1 x_2 - x_2 y_2 b}{x_1 y_1}.$$

We are interested only in integer solutions, that is when a, c, and b are in $\mathbb{Z}$, which means that b must satisfy the following congruences:

$$\begin{aligned} x_2 b &\equiv y_1 \mod x_1 \\ y_2 b &\equiv -x_1 \mod y_1 \\ x_2 y_2 b &\equiv y_1 y_2 - x_1 x_2 \mod x_1 y_1 \end{aligned}$$

Applying the algorithm from Lemma 5 two times, we can determine in PTIME whether the above system has a solution or not. If the solution exists, the algorithm outputs it in the form $b \equiv b_2 \mod b_1$, where $b_1 \mid x_1 y_1$.

So, the coefficient b is of the form $b = b_1 t + b_2$, where $t \in \mathbb{Z}$. Substituting this expression for b in the formulas for a, c, and d we obtain:

$$a = \frac{y_1 - x_2 b_2 - x_2 b_1 t}{x_1} = a_1 t + a_2,$$
$$d = \frac{x_1 + y_2 b_2 + y_2 b_1 t}{y_1} = d_1 t + d_2,$$
$$c = \frac{y_1 y_2 - x_1 x_2 - x_2 y_2 b_2 - x_2 y_2 b_1 t}{x_1 y_1} = c_1 t + c_2,$$

where a_i, c_i, and d_i, for $i = 1, 2$, are some constants which are necessarily in $\mathbb{Z}$ because if we let $t = 0$ or $t = 1$ in the above expressions they must evaluate to integer numbers. Therefore, the solution to the system of equations (2)–(4) can be written as:

$$M = \begin{bmatrix} a_1 t + a_2 & b_1 t + b_2 \\ c_1 t + c_2 & d_1 t + d_2 \end{bmatrix} = t \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} + \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix},$$

where t is any integer number. To complete the proof we set $A_1 = \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix}$ and $A_2 = \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix}$. Note that the above algorithm runs in polynomial time because the only nontrivial step is to solve the system of linear congruence equations, which according to Lemma 5 can be done in PTIME. ◀

For the next proposition we will need the following theorem about the Smith normal form of a matrix.

▶ **Theorem 7** (Smith normal form [15]). *For any non-zero matrix $A \in \mathbb{Z}^{2\times 2}$, there are matrices $B, C \in \mathrm{SL}(2, \mathbb{Z})$ such that*

$$A = B \begin{bmatrix} t_1 & 0 \\ 0 & t_2 \end{bmatrix} C$$

for some $t_1, t_2 \in \mathbb{Z}$ such that $t_1 \neq 0$ and $t_1 \mid t_2$. Moreover, B, C, t_1, t_2 can be computed in PTIME.

▶ **Proposition 8.** *Let A_1 and A_2 be matrices from $\mathbb{Z}^{2\times 2}$ such that for every $t \in \mathbb{Z}$, we have $tA_1 + A_2 \in \mathrm{SL}(2, \mathbb{Z})$. Then there are matrices B and C from $\mathrm{SL}(2, \mathbb{Z})$ and $k \in \mathbb{Z}$ such that*

$$tA_1 + A_2 = BT^{kt}C \quad \text{for every } t \in \mathbb{Z},$$

where $T = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \in \mathrm{SL}(2, \mathbb{Z})$. Moreover, B, C, and k can be computed in PTIME.

Proof. Let $A_1 = \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix}$ and $A_2 = \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix}$. By assumption, for every $t \in \mathbb{Z}$, $\begin{vmatrix} a_1 t + a_2 & b_1 t + b_2 \\ c_1 t + c_2 & d_1 t + d_2 \end{vmatrix} = 1$. That is

$$(a_1 t + a_2)(d_1 t + d_2) - (b_1 t + b_2)(c_1 t + c_2) = 1 \quad \text{or}$$
$$(a_1 d_1 - b_1 c_1) t^2 + (a_1 d_2 + a_2 d_1 - b_1 c_2 - b_2 c_1) t + a_2 d_2 - b_2 c_2 = 1 \quad \text{for all } t \in \mathbb{Z}.$$

Therefore, $a_1d_1 - b_1c_1 = 0$, $a_1d_2 + a_2d_1 - b_1c_2 - b_2c_1 = 0$, and $a_2d_2 - b_2c_2 = 1$. In particular, $\det(A_1) = 0$ and $\det(A_2) = 1$.

If A_1 is zero matrix, then the proof is trivial. So suppose A_1 is a non-zero matrix. Then by Theorem 7, there are matrices $F, G \in \mathrm{SL}(2,\mathbb{Z})$ such that $A_1 = F\begin{bmatrix} k & 0 \\ 0 & 0 \end{bmatrix} G$ for some $k \in \mathbb{Z} \setminus \{0\}$.

Now $F^{-1}(tA_1 + A_2)G^{-1} = \begin{bmatrix} kt + a & b \\ c & d \end{bmatrix}$, for some $a, b, c, d \in \mathbb{Z}$. Note that since $\det(F) = \det(G) = \det(tA_1 + A_2) = 1$, we have

$$\begin{vmatrix} kt + a & b \\ c & d \end{vmatrix} = dkt + ad - bc = 1 \quad \text{for every } t \in \mathbb{Z}. \tag{5}$$

Hence $dk = 0$ and so $d = 0$. Substituting $d = 0$ in (5), we get $bc = -1$. Since b and c are integers, there are only two possibilities: $b = 1$, $c = -1$, or $b = -1$, $c = 1$. So the above matrix actually looks like

$$F^{-1}(tA_1 + A_2)G^{-1} = \begin{bmatrix} kt + a & \mp 1 \\ \pm 1 & 0 \end{bmatrix}.$$

Therefore, $T^{-c(kt+a)}F^{-1}(tA_1 + A_2)G^{-1} = D$, where $c = \pm 1$ and $D = \begin{bmatrix} 0 & \mp 1 \\ \pm 1 & 0 \end{bmatrix} \in \mathrm{SL}(2,\mathbb{Z})$. Hence

$$tA_1 + A_2 = FT^{(ck)t}T^{ca}DG.$$

Note that F and $T^{ca}DG$ are in $\mathrm{SL}(2,\mathbb{Z})$. This completes the proof. The bound on complexity follows from the fact that F and G can be computed in PTIME by Theorem 7. ◀

As a corollary of Propositions 6 and 8 we obtain the following theorem.

▶ **Theorem 9.** *Let* $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ *and* $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ *be vectors from* $\mathbb{Z} \times \mathbb{Z}$ *and consider the matrix equation* $M\mathbf{x} = \mathbf{y}$*, where* M *is an unknown matrix from* $\mathrm{SL}(2,\mathbb{Z})$*. Then either this equation does not have a solution or all its solutions are given by the following formula*

$$M = B\begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}^t C, \quad \text{where } t \in \mathbb{Z}.$$

In the above expression B *and* C *are some matrices from* $\mathrm{SL}(2,\mathbb{Z})$*, and* k *is an integer number. Moreover, there is a polynomial time algorithm that determines whether such equation has a solution and if so, finds the suitable matrices* B*,* C *and the integer* k*.*

Now we turn to the reachability problem by fractional linear transformations.

▶ **Proposition 10.** *Let* x *and* y *be rational numbers and let* $\mathcal{F}(x,y)$ *be the following set of matrices from* $\mathrm{SL}(2,\mathbb{Z})$

$$\mathcal{F}(x,y) = \{M \in \mathrm{SL}(2,\mathbb{Z}) \ : \ f_M(x) = y\}.$$

Then either $\mathcal{F}(x,y)$ *is empty or there is a finite collection of matrices* $A_1, \ldots, A_n$ *and* $B_1, \ldots, B_n$ *from* $\mathbb{Z}^{2\times 2}$ *such that*

$$\mathcal{F}(x,y) = \bigcup_{i=1}^{n} \{A_i t + B_i \ : \ t \in \mathbb{Z}\}.$$

Moreover, the matrices $A_1, \ldots, A_n$ *and* $B_1, \ldots, B_n$ *can be computed effectively from* x *and* y*.*

Proof. Let $x = \frac{m_0}{n_0}$ and $y = \frac{m_1}{n_1}$, where $m_0, m_1 \in \mathbb{Z}$ and $n_0, n_1 \in \mathbb{N}^+$ and $\gcd(m_0, n_0) = \gcd(m_1, n_1) = 1$. Let $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ be a matrix from $\mathrm{SL}(2, \mathbb{Z})$ such that $f_M(x) = y$. Then we have the following system of equations in which a, b, c, d are unknowns

$$\frac{a\frac{m_0}{n_0} + b}{c\frac{m_0}{n_0} + d} = \frac{m_1}{n_1} \quad \text{and} \quad ad - bc = 1.$$

Let us choose c, d as free parameters and express a, b in terms of them. We have

$$\frac{am_0 + bn_0}{cm_0 + dn_0} = \frac{m_1}{n_1} \quad \text{or} \quad am_0n_1 + bn_0n_1 = cm_0m_1 + dn_0m_1.$$

Since $n_0n_1 \neq 0$ we can write

$$b = \frac{1}{n_0n_1}(-am_0n_1 + cm_0m_1 + dn_0m_1).$$

Substituting this expression for b into the equation $ad - bc = 1$ gives us

$$\begin{aligned} ad - \frac{c}{n_0n_1}(-am_0n_1 + cm_0m_1 + dn_0m_1) &= 1, \\ adn_0n_1 + acm_0n_1 - c^2m_0m_1 - cdn_0m_1 &= n_0n_1, \\ an_1(cm_0 + dn_0) &= n_0n_1 + cm_1(cm_0 + dn_0). \end{aligned}$$

Note that in the above equation $cm_0 + dn_0 \neq 0$ since otherwise we would have a contradiction $n_0n_1 = 0$. Thus we have

$$\begin{aligned} a &= \frac{cm_1}{n_1} + \frac{n_0}{cm_0 + dn_0} \quad \text{and} \\ b &= \frac{1}{n_0n_1}(-cm_0m_1 - \frac{n_0m_0n_1}{cm_0 + dn_0} + cm_0m_1 + dn_0m_1) = \frac{dm_1}{n_1} - \frac{m_0}{cm_0 + dn_0}. \end{aligned}$$

We want to find all integer values of c, d for which the corresponding values of a, b are also integers. Looking at the expressions for a and b above, one can notice that they both are equal to sums of two fractions with denominators n_1 and $cm_0 + dn_0$, respectively. Now observe that the fractional part of $\frac{cm_1}{n_1}$ depends only on the residue of c modulo n_1. Similarly, the fractional part of $\frac{dm_1}{n_1}$ depends only on the residue of d modulo n_1. Therefore, we need to analyze n_1^2 many cases when $c \equiv i \mod n_1$ and $d \equiv j \mod n_1$ for all $i, j \in \{0, \ldots, n_1 - 1\}$. Hence we will have $\mathcal{F}(x, y) = \bigcup_{i,j \in \{0,\ldots,n_1-1\}} \mathcal{F}_{i,j}$, where

$$\mathcal{F}_{i,j} \;=\; \left\{ M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathrm{SL}(2, \mathbb{Z}) \;:\; c \equiv i \mod n_1,\; d \equiv j \mod n_1, \text{ and } f_M(x) = y \right\}.$$

So, suppose that c and d are of the form $c = i + n_1k$ and $d = j + n_1l$, where $k, l \in \mathbb{Z}$. In this case

$$\begin{aligned} a &= m_1k + \frac{m_1i}{n_1} + \frac{n_0}{m_0i + m_0n_1k + n_0j + n_0n_1l} \quad \text{and} \\ b &= m_1l + \frac{m_1j}{n_1} - \frac{m_0}{m_0i + m_0n_1k + n_0j + n_0n_1l}. \end{aligned} \tag{6}$$

We want to find all values of $k, l \in \mathbb{Z}$ for which the corresponding values of a, b are integer. First, consider an expression $\frac{m_1i}{n_1} + \frac{n_0}{x}$, where x is an unknown variable. Note that there are only finitely many possible integer values for x for which the above expression evaluates

to an integer number. For instance, if $\frac{m_1 i}{n_1}$ is an integer, then x must be a divisor of n_0, in particular, $|x| \leq n_0$. On the other hand, if $\frac{m_1 i}{n_1}$ is a proper fraction, then $\frac{n_0}{|x|} \geq \frac{1}{n_1}$ since otherwise $\frac{m_1 i}{n_1} + \frac{n_0}{x}$ cannot be an integer. Thus we have $|x| \leq n_0 n_1$. Therefore, there are only finitely many integer values of x for with *both* expressions $\frac{m_1 i}{n_1} + \frac{n_0}{x}$ and $\frac{m_1 j}{n_1} - \frac{m_0}{x}$ are integers.

Let $X_{i,j} = \{x_1, \ldots, x_s\}$ be the set of all such values of x. If $X_{i,j} = \emptyset$, then we have $\mathcal{F}_{i,j} = \emptyset$. Otherwise, we will have $\mathcal{F}_{i,j} = \bigcup_{r=1}^{s} \mathcal{F}^r_{i,j}$, where

$$\mathcal{F}^r_{i,j} = \left\{ M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathrm{SL}(2,\mathbb{Z}) \ : c \equiv i \mod n_1,\ d \equiv j \mod n_1, \right.$$
$$\left. cm_0 + dn_0 = x_r, \text{ and } f_M(x) = y \right\}.$$

To compute $\mathcal{F}^r_{i,j}$, consider the following equation in which k, l are unknown variables

$$m_0 i + m_0 n_1 k + n_0 j + n_0 n_1 l = x_r.$$

We want to find all integer values of k, l which satisfy this equation. Note that since $n_0 n_1 \neq 0$ we can express l as a linear function of k:

$$l = \frac{x_r - m_0 i - n_0 j - m_0 n_1 k}{n_0 n_1}. \tag{7}$$

The values of k for which the corresponding value of l is integer must satisfy the following modular equation

$$m_0 n_1 k \equiv x_r - m_0 i - n_0 j \mod n_0 n_1.$$

By Lemma 4, such equation either has no solution or it has a unique solution of the form $k \equiv k_1 \mod k_0$ for some $k_0 \mid n_0 n_1$. In case when the above equation has no solution, we have $\mathcal{F}^r_{i,j} = \emptyset$. So suppose there is a unique solution, which we can rewrite as $k = k_0 t + k_1$, where $t \in \mathbb{Z}$. From (7) we obtain that $l = l_0 t + l_1$, where

$$l_0 = -\frac{m_0 k_0}{n_0} \quad \text{and} \quad l_1 = \frac{x_r - m_0 i - n_0 j - m_0 n_1 k_1}{n_0 n_1}.$$

Note that both l_0 and l_1 are integer numbers because, by our construction, the value of l must be integer for all values of $t \in \mathbb{Z}$. Now we have

$$c = i + n_1 k = i + n_1 k_1 + n_1 k_0 t = c_0 t + c_1, \quad \text{and}$$
$$d = j + n_1 l = j + n_1 l_1 + n_1 l_0 t = d_0 t + d_1,$$

where $c_0 = n_1 k_0$, $c_1 = i + n_1 k_1$, $d_0 = n_1 l_0$, and $d_1 = j + n_1 l_1$. Furthermore, from (6) we obtain

$$a = m_1 k + \frac{m_1 i}{n_1} + \frac{n_0}{x_r} = m_1 k_0 t + m_1 k_1 + \frac{m_1 i}{n_1} + \frac{n_0}{x_r} = a_0 t + a_1 \quad \text{and}$$
$$b = m_1 l + \frac{m_1 j}{n_1} - \frac{m_0}{x_r} = m_1 l_0 t + m_1 l_1 + \frac{m_1 j}{n_1} - \frac{m_0}{x_r} = b_0 t + b_1,$$

where $a_0 = m_1 k_0$, $a_1 = m_1 k_1 + \frac{m_1 i}{n_1} + \frac{n_0}{x_r}$, $b_0 = m_1 l_0$, and $b_1 = m_1 l_1 + \frac{m_1 j}{n_1} - \frac{m_0}{x_r}$. Note that a_1 and b_1 are integers by the choice of x_r. Finally, we have

$$\mathcal{F}^r_{i,j} = \{At + B \ : \ t \in \mathbb{Z}\}, \quad \text{where } A = \begin{bmatrix} a_0 & b_0 \\ c_0 & d_0 \end{bmatrix} \text{ and } B = \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix}.$$

Thus we have shown that $\mathcal{F}(x,y) = \bigcup_{i,j\in\{0,\ldots,n_1-1\}} \mathcal{F}_{i,j}$, where every $\mathcal{F}_{i,j}$ is either empty or has the form $\mathcal{F}_{i,j} = \bigcup_{r=1}^{s} \mathcal{F}_{i,j}^r$, where $\mathcal{F}_{i,j}^r$ is either empty or has the form $\mathcal{F}_{i,j}^r = \{At+B \ : \ t \in \mathbb{Z}\}$ for some matrices A and B from $\mathbb{Z}^{2\times 2}$.

It is not hard to see that the procedure described above is effective. ◀

Combining Proposition 10 and Proposition 8 we obtain the following theorem.

▶ **Theorem 11.** *Let x and y be rational numbers and let $\mathcal{F}(x,y)$ be the following set of matrices from* $\mathrm{SL}(2,\mathbb{Z})$

$$\mathcal{F}(x,y) = \{M \in \mathrm{SL}(2,\mathbb{Z}) \ : \ f_M(x) = y\}.$$

Then either $\mathcal{F}(x,y)$ is empty or there is a finite collection of matrices $C_1,\ldots,C_n$ and $D_1,\ldots,D_n$ from $\mathrm{SL}(2,\mathbb{Z})$ *and integers $s_1,\ldots,s_n \in \mathbb{Z}$ such that*

$$\mathcal{F}(x,y) = \bigcup_{i=1}^{n} \left\{ C_i \begin{bmatrix} 1 & s_i \\ 0 & 1 \end{bmatrix}^t D_i \ : \ t \in \mathbb{Z} \right\}.$$

Moreover, the matrices $C_1,\ldots,C_n$, $D_1,\ldots,D_n$ and integers $s_1,\ldots,s_n$ can be computed effectively from x and y.

Now we prove that the emptiness problem for the intersection of two regular subsets of $\mathrm{SL}(2,\mathbb{Z})$ is decidable.

Consider an alphabet $\Sigma = \{S,R\}$ consisting of two symbols S and R and define the mapping $\varphi : \Sigma \to \mathrm{SL}(2,\mathbb{Z})$ as follows: $\varphi(S) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ and $\varphi(R) = \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}$. We can extend this mapping to the morphism $\varphi : \Sigma^* \to \mathrm{SL}(2,\mathbb{Z})$ in the usual way. Note that $\varphi(S)$ and $\varphi(R)$ are generators of $\mathrm{SL}(2,\mathbb{Z})$ [18], so φ is surjective.

▶ **Definition 12.** A *signed automaton* is a (non-deterministic) finite automaton $A = (\Sigma, Q, I, \Delta, F^+, F^-)$ whose final states are divided into two (not necessarily disjoint) subsets F^+ and F^-.

A *signed language* accepted by a signed automaton A is a pair $L(A) = (L(A)^+, L(A)^-)$, where $L(A)^+$ and $L(A)^-$ consists of the words $w \in \Sigma^*$ for which there is a run of A that ends in the set F^+ or F^-, respectively. Note that we do not assume that $L(A)^+$ and $L(A)^-$ are disjoint.

If $L = (L^+, L^-)$ is a signed language, then we define

$$\varphi(L) = \{\varphi(w) \ : \ w \in L^+\} \cup \{-\varphi(w) \ : \ w \in L^-\}.$$

The following proposition is an important ingredient of our main results.

▶ **Proposition 13.** *There is an algorithm that for any given regular signed languages L_1 and L_2 over the alphabet Σ, decides whether $\varphi(L_1) \cap \varphi(L_2)$ is empty or not.*

Proof. We call a word $w \in \Sigma^*$ *reduced* if it does not have substrings of the form SS or RRR. Note that for every $M \in \mathrm{SL}(2,\mathbb{Z})$, there is a unique reduced word $w \in \Sigma^*$ such that either $M = \varphi(w)$ or $M = -\varphi(w)$ [17, 18].

We now describe a construction that turns any signed automaton A over Σ into a new signed automaton $\widetilde{A}$ such that

- $\varphi(L(\widetilde{A})) = \varphi(L(A))$ and

- for every $M \in \varphi(L(\widetilde{A}))$, there is a reduced word w such that $M = \varphi(w)$ or $M = -\varphi(w)$ and $w \in L(\widetilde{A})^+$ or $w \in L(\widetilde{A})^-$, respectively.

Suppose $A = (\Sigma, Q, I, \Delta, F^+, F^-)$, then $\widetilde{A}$ is defined as follows $\widetilde{A} = (\Sigma, \widetilde{Q}, \widetilde{I}, \widetilde{\Delta}, \widetilde{F}^+, \widetilde{F}^-)$, where

- $\widetilde{Q} = Q \times \{+, -\}$,
- $\widetilde{I} = I \times \{+\}$,
- $\widetilde{F}^+ = \{(q, +) : q \in F^+\} \cup \{(q, -) : q \in F^-\}$,
- $\widetilde{F}^- = \{(q, +) : q \in F^-\} \cup \{(q, -) : q \in F^+\}$,

For each transition $(q_1, X, q_2) \in \Delta$, we add the following two transition in $\widetilde{\Delta}$: $((q_1, +), X, (q_2, +))$ and $((q_1, -), X, (q_2, -))$.

Furthermore, we iteratively add new ε-transitions to $\widetilde{\Delta}$ as follows: if there is a run of $\widetilde{A}$ from (q_1, s_1) to (q_2, s_2) labelled by SS or RRR, then we add an ε-transition from (q_1, s_1) to $(q_2, \bar{s}_2)$, where $\bar{s}_2$ is the sign opposite to s_2. For instance, if there is a run from $(q_1, +)$ to $(q_2, +)$ labelled by RRR, then we add an ε-transition from $(q_1, +)$ to $(q_2, -)$ (see Figure 1 in the Appendix for an illustration). We continue this process until no new ε-transitions can be added.

Note that in $\mathrm{SL}(2, \mathbb{Z})$ we have $\varphi(S)^2 = \varphi(R)^3 = -I$, and this is reflected in the change of sign of the end state of a new ε-transition. It is not hard to see that $\widetilde{A}$ is indeed the desired automaton.

Let A_1 and A_2 be two finite signed automata such that $L(A_1) = L_1$ and $L(A_2) = L_2$. To check whether $\varphi(L_1) \cap \varphi(L_2)$ is empty or not, we take the automata A_1 and A_2 and construct the new automata $\widetilde{A}_1$ and $\widetilde{A}_2$ as described above.

Now we have

$$\varphi(L_1) \cap \varphi(L_2) \neq \emptyset \text{ if and only if } L(\widetilde{A}_1)^+ \cap L(\widetilde{A}_2)^+ \neq \emptyset \text{ or } L(\widetilde{A}_1)^- \cap L(\widetilde{A}_2)^- \neq \emptyset.$$

Indeed, suppose that $M \in \varphi(L_1) \cap \varphi(L_2)$. By the above construction we have $\varphi(L(\widetilde{A}_i)) = \varphi(L_i)$, for $i = 1, 2$, and there is a reduced word $w \in \Sigma^*$ such that $M = \varphi(w)$ or $M = -\varphi(w)$ and $w \in L(\widetilde{A}_i)^+$ or $w \in L(\widetilde{A}_i)^-$, respectively, for both $i = 1, 2$. In the fist case we have $w \in L(\widetilde{A}_1)^+ \cap L(\widetilde{A}_2)^+$ and in the second case $w \in L(\widetilde{A}_1)^- \cap L(\widetilde{A}_2)^-$. The implication in the other direction is trivial.

To complete the proof we note that the intersection of regular languages is again regular, and the emptiness problem for regular languages is decidable. ◀

We are now ready to prove our main results.

▶ **Theorem 14.** *The vector reachability problem in* $\mathrm{SL}(2, \mathbb{Z})$ *is decidable.*

▶ **Theorem 15.** *The reachability problem by fractional linear transformations in* $\mathrm{SL}(2, \mathbb{Z})$ *is decidable.*

Proof of Theorems 14 and 15. Suppose that we are given a finite collection of matrices $M_1, \ldots, M_n$ from $\mathrm{SL}(2, \mathbb{Z})$. Let $w_1, \ldots, w_n \in \Sigma^*$ be some words, not necessarily reduced, such that $M_i = \varphi(w_i)$, for $i = 1, \ldots, n$. Define the language $\mathcal{L}_{semigr}$ that corresponds to the semigroup $\langle M_1, \ldots, M_n \rangle$ as $\mathcal{L}_{semigr} = (w_1 + w_2 + \cdots + w_n)^*$.

Recall that in the vector reachability problem we are given two vectors $\mathbf{x}$ and $\mathbf{y}$ from $\mathbb{Z} \times \mathbb{Z}$, and we ask if there is a matrix $M \in \langle M_1, \ldots, M_n \rangle$ such that $M\mathbf{x} = \mathbf{y}$. We want to construct a regular language $\mathcal{L}^{\mathrm{vrp}}_{\mathbf{x},\mathbf{y}}$ that corresponds to this problem. By Theorem 9, the matrix equation $M\mathbf{x} = \mathbf{y}$ either has no solution, or its solution is equal to $\{BT^{kt}C \ : \ t \in \mathbb{Z}\}$, where $T = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, B and C are some matrices from $\mathrm{SL}(2, \mathbb{Z})$, and k is an integer number.

Moreover, B, C and k can be effectively computed from $\mathbf{x}$ and $\mathbf{y}$. Without loss of generality, we can assume that $k \geq 0$ since we can replace k with $-k$ if necessary.

In the case when $M\mathbf{x} = \mathbf{y}$ has no solution, we set $\mathcal{L}^{\text{vrp}}_{\mathbf{x},\mathbf{y}} = \emptyset$. Suppose that the solution set in non-empty. In this case we can rewrite it as

$$\{BT^{kt}C \,:\, t \in \mathbb{Z}\} = \{B(T^k)^tC \,:\, t \geq 0\} \cup \{B(T^{-k})^tC \,:\, t \geq 0\}.$$

Let u_1 and u_2 be words from Σ^* such that $B = \varphi(u_1)$ and $C = \varphi(u_2)$. It is easy to check that $T = \varphi(S^3R)$ and $T^{-1} = \varphi(R^5S)$. Hence

$$\mathcal{L}^{\text{vrp}}_{\mathbf{x},\mathbf{y}} = u_1\big((S^3R)^k\big)^*u_2 + u_1\big((R^5S)^k\big)^*u_2$$

is a regular language that describes the solutions of the equation $M\mathbf{x} = \mathbf{y}$ in $\mathrm{SL}(2,\mathbb{Z})$.

In a similar way we can construct a regular language $\mathcal{L}^{\text{flt}}_{x,y}$ that corresponds to the reachability problem by fractional linear transformations from x to y. By Theorem 11, the set $\mathcal{F}(x,y)$ of matrices from $\mathrm{SL}(2,\mathbb{Z})$ that satisfy the equation $f_M(x) = y$ is either empty or has the form

$$\mathcal{F}(x,y) = \bigcup_{i=1}^{n}\{C_iT^{s_it}D_i \,:\, t \in \mathbb{Z}\} = \bigcup_{i=1}^{n}\{C_iT^{s_it}D_i \,:\, t \geq 0\} \cup \bigcup_{i=1}^{n}\{C_i(T^{-s_i})^tD_i \,:\, t \geq 0\},$$

where T is as above, C_i and D_i are some matrices from $\mathrm{SL}(2,\mathbb{Z})$, and s_i are integer numbers. All these matrices and numbers can be effectively computed from x and y. Again we can assume that $s_i \geq 0$, for $i = 1,\ldots,n$.

If $\mathcal{F}(x,y) = \emptyset$, then we set $\mathcal{L}^{\text{flt}}_{x,y} = \emptyset$. Otherwise, let u_i and v_i be words from Σ^* such that $C_i = \varphi(u_i)$ and $D_i = \varphi(v_i)$, for $i = 1,\ldots,n$. Let

$$\mathcal{L}^{\text{flt}}_{x,y} = \bigcup_{i=1}^{n} u_i\big((S^3R)^{s_i}\big)^*v_i + \bigcup_{i=1}^{n} u_i\big((R^5S)^{s_i}\big)^*v_i.$$

Then $\mathcal{L}^{\text{flt}}_{x,y}$ is a regular language the describes the solution of the equation $f_M(x) = y$ in $\mathrm{SL}(2,\mathbb{Z})$. We remind that in Proposition 13 we are working with signed languages. Therefore, in what follows we convert every regular language L that we have constructed so far into a corresponding signed language $(L,\emptyset)$.

Finally, the vector reachability problem for $\mathbf{x}$ and $\mathbf{y}$ has a solution if and only if $\varphi\big((\mathcal{L}^{\text{vrp}}_{\mathbf{x},\mathbf{y}},\emptyset)\big) \cap \varphi\big((\mathcal{L}_{semigr},\emptyset)\big) \neq \emptyset$. Similarly, the reachability problem by fractional linear transformations for x and y has a solution if and only if $\varphi\big((\mathcal{L}^{\text{flt}}_{x,y},\emptyset)\big) \cap \varphi\big((\mathcal{L}_{semigr},\emptyset)\big) \neq \emptyset$. By Proposition 13 these questions are algorithmically decidable. ◀

Conclusion and future work Apart from solving two open problems for matrix semigroups the results of this paper have further consequences and can be extended in several ways. In the solution of the vector reachability problem for given two vectors $\mathbf{x},\mathbf{y} \in \mathbb{Z}^2$ we characterize linear transformation from $\mathrm{SL}(2,\mathbb{Z})$ that map $\mathbf{x}$ to $\mathbf{y}$ and express them both in a matrix and symbolic forms as a regular expression that can be computed in polynomial time. The proposed algorithm is currently EXPSPACE, due to the fact that the exponential explosion happens after construction of $\mathcal{L}_{semigr}$. However the PTIME algorithm for computing a mapping from $\mathbf{x}$ to $\mathbf{y}$ could be combined with the result of Gurevich and Schupp [12] to produce a polynomial time algorithm for the vector reachability for the modular group. Moreover any improvement of EXPSPACE solution proposed in [9] will improve the complexity of the vector reachability problem. In addition we believe that our proof for the decidability of the vector reachability problem in $\mathrm{SL}(2,\mathbb{Z})$ can be extended from $\mathbb{Z}^2$ to complex numbers with rational coordinates, and the solution for linear fractional transformations could be used for solving a similar problem in the context of deterministic piecewise iterative functions.

References

1 László Babai, Robert Beals, Jin-yi Cai, Gábor Ivanyos, and Eugene M. Luks. Multiplicative equations over commuting matrices. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '96, pages 498–507, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.

2 Paul Bell and Igor Potapov. On undecidability bounds for matrix decision problems. *Theoretical Computer Science*, 391(1-2):3–13, 2008.

3 Paul Bell and Igor Potapov. Reachability problems in quaternion matrix and rotation semigroups. *Information and Computation*, 206(11):1353–1361, 2008.

4 Paul C. Bell, Mika Hirvensalo, and Igor Potapov. Mortality for 2x2 matrices is NP-hard. In Branislav Rovan, Vladimiro Sassone, and Peter Widmayer, editors, *Mathematical Foundations of Computer Science 2012*, volume 7464 of *Lecture Notes in Computer Science*, pages 148–159. Springer Berlin Heidelberg, 2012.

5 Paul C. Bell and Igor Potapov. On the computational complexity of matrix semigroup problems. *Fundam. Inf.*, 116(1-4):1–13, January 2012.

6 Vincent D. Blondel, Emmanuel Jeandel, Pascal Koiran, and Natacha Portier. Decidable and undecidable problems about quantum automata. *SIAM J. Comput.*, 34(6):1464–1473, June 2005.

7 Julien Cassaigne, Tero Harju, and Juhani Karhumaki. On the undecidability of freeness of matrix semigroups. *International Journal of Algebra and Computation*, 09(03n04):295–305, 1999.

8 Julien Cassaigne and François Nicolas. On the decidability of semigroup freeness. *RAIRO - Theor. Inf. and Applic.*, 46(3):355–399, 2012.

9 Choffrut, Christian and Karhumaki, Juhani. Some decision problems on integer matrices. *RAIRO-Theor. Inf. Appl.*, 39(1):125–131, 2005.

10 J. Esparza, A. Finkel, and R. Mayr. On the verification of broadcast protocols. In *Logic in Computer Science, 1999. Proceedings. 14th Symposium on*, pages 352–359, 1999.

11 Esther Galby, Joël Ouaknine, and James Worrell. On Matrix Powering in Low Dimensions. In Ernst W. Mayr and Nicolas Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 329–340, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

12 Yuri Gurevich and Paul Schupp. Membership problem for the modular group. *SIAM J. Comput.*, 37(2):425–459, May 2007.

13 Vesa Halava, Tero Harju, and Mika Hirvensalo. Undecidability bounds for integer matrices using Claus instances. *International Journal of Foundations of Computer Science*, 18(05):931–948, 2007.

14 Vesa Halava, Tero Harju, Mika Hirvensalo, and Juhani Karhumaki. Skolem's problem - on the border between decidability and undecidability. Technical Report 683, Turku Centre for Computer Science, 2005.

15 Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8(4):499–507, 1979.

16 Alexei Lisitsa and Igor Potapov. Membership and reachability problems for row-monomial transformations. In *Mathematical Foundations of Computer Science 2004, 29th International Symposium, MFCS 2004, Prague, Czech Republic, August 22-27, 2004, Proceedings*, pages 623–634, 2004.

17 Roger C. Lyndon and Paul E. Schupp. *Combinatorial group theory*. Springer-Verlag, Berlin-New York, 1977. Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 89.

18 Wilhelm Magnus, Abraham Karrass, and Donald Solitar. *Combinatorial group theory.* Dover Publications, Inc., New York, revised edition, 1976. Presentations of groups in terms of generators and relations.

19 A. Markov. On certain insoluble problems concerning matrices. *Doklady Akad. Nauk SSSR*, 57(6):539–542, June 1947.

20 Joël Ouaknine, João Sousa Pinto, and James Worrell. On termination of integer linear loops. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 957–969. SIAM, 2015.

21 Joël Ouaknine and James Worrell. On the positivity problem for simple linear recurrence sequences,. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, pages 318–329, 2014.

22 Joël Ouaknine and James Worrell. Ultimate positivity is decidable for simple linear recurrence sequences. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, pages 330–341, 2014.

A Appendix

▶ **Lemma 4.** *Consider a linear congruence equation* $ax \equiv b \mod n$. *If* $\gcd(a,n) \nmid b$, *then the equation has no solution. If* $\gcd(a,n) \mid b$, *then all solutions of the equation can be written in the form* $x \equiv c \mod \frac{n}{\gcd(a,n)}$ *for some* c. *Moreover, there is a polynomial time algorithm that determines whether such equation has a solution and if so, finds it.*

Proof. Given a and n, using Euclidean algorithm we can find in polynomial time $d = \gcd(a,n)$ and integer numbers u and v such that $d = ua + vn$. Equation $ax \equiv b \mod n$ can be written as $ax = b + kn$, where $k \in \mathbb{Z}$. It is clear that if $d \nmid b$, then there is no solution. Otherwise, let $b = b'd$, $a = a'd$, and $n = n'd$. Then our equation is equivalent to $a'x \equiv b' \mod n'$. Furthermore, we have $ua' + vn' = 1$ and hence $ua' \equiv 1 \mod n'$. Thus

$$x \equiv (ua')x \equiv u(a'x) \equiv ub' \mod n'.$$

Note that all these computations can be done in PTIME. ◀

▶ **Lemma 5.** *Consider a system of two linear congruence equations*

$$\begin{aligned} a_1 x &\equiv b_1 \mod n_1 \\ a_2 x &\equiv b_2 \mod n_2 \end{aligned} \tag{8}$$

Such system either has no solution, or all its solutions are of the form $x \equiv c \mod n$ *for some* c *and* $n \mid n_1 n_2$. *Moreover, there is a polynomial time algorithm that determines whether (8) has a solution and if so, finds it.*

Proof. Using the algorithm of Lemma 4, we can solve each equation separately. If one of them does not have a solution, then the system (8) also has no solution. Suppose the first and second equation have the solutions $x \equiv c_1 \mod n_1'$ and $x \equiv c_2 \mod n_2'$, respectively, which can be found in PTIME. Note that $n_i' \mid n_i$ for $i = 1, 2$.

Let $n = \operatorname{lcm}(n_1', n_2')$. We can rewrite the solutions as

$$\begin{aligned} x &\equiv c_1,\ c_1 + n_1',\ c_1 + 2n_1',\ \ldots,\ c_1 + (n_2'' - 1)n_1' \mod n, \\ x &\equiv c_2,\ c_2 + n_2',\ c_2 + 2n_2',\ \ldots,\ c_2 + (n_1'' - 1)n_2' \mod n, \end{aligned}$$

where $n_1'' = n/n_2'$ and $n_2'' = n/n_1'$. Let

$$A_1 = \{c_1,\ c_1 + n_1',\ c_1 + 2n_1',\ \ldots,\ c_1 + (n_2'' - 1)n_1'\},$$
$$A_2 = \{c_2,\ c_2 + n_2',\ c_2 + 2n_2',\ \ldots,\ c_2 + (n_1'' - 1)n_2'\}.$$

Note that $A_1 \cap A_2$ contains at most one element. Indeed, if $c, c' \in A_1 \cap A_2$, then $n_1' \mid c - c'$ and $n_2' \mid c - c'$. Hence $n = \mathrm{lcm}(n_1', n_2') \mid c - c'$. Since $|c - c'| < n$, we have $c = c'$.

Now if $A_1 \cap A_2$ is empty, then (8) has no solution. If $A_1 \cap A_2 = \{c\}$, then the solution of (8) is $x \equiv c \mod n$. To find this solution in PTIME, observe the following. The equations $x \equiv c_1 \mod n_1'$ and $x \equiv c_2 \mod n_2'$ are equivalent to $x = c_1 + kn_1'$ and $x = c_2 + ln_2'$, respectively, where $k, l \in \mathbb{Z}$. To find the intersection of these solutions we set $c_1 + kn_1' = c_2 + ln_2'$, which is equivalent to $c_1 - c_2 = ln_2' - kn_1'$. Using Euclidean algorithm, we can find in PTIME $d = \gcd(n_1', n_2')$ and integer numbers u, v such that

$$d = un_1' + vn_2'. \tag{9}$$

Obviously, if $d \nmid c_1 - c_2$, then there is no solution. So suppose $c_1 - c_2 = hd$, for some $h \in \mathbb{Z}$. Multiplying (9) by h we obtain

$$c_1 - c_2 = hd = (hu)n_1' + (hv)n_2' \quad \text{or}$$
$$c_1 - (hu)n_1' = c_2 + (hv)n_2'.$$

Let c be the number in the set $\{0, \ldots, n-1\}$ such that

$$c \equiv c_1 - (hu)n_1' = c_2 + (hv)n_2' \mod n.$$

Then $x \equiv c \mod n$ is the desired solution. It is not hard to see that the above algorithm runs in polynomial time. ◀

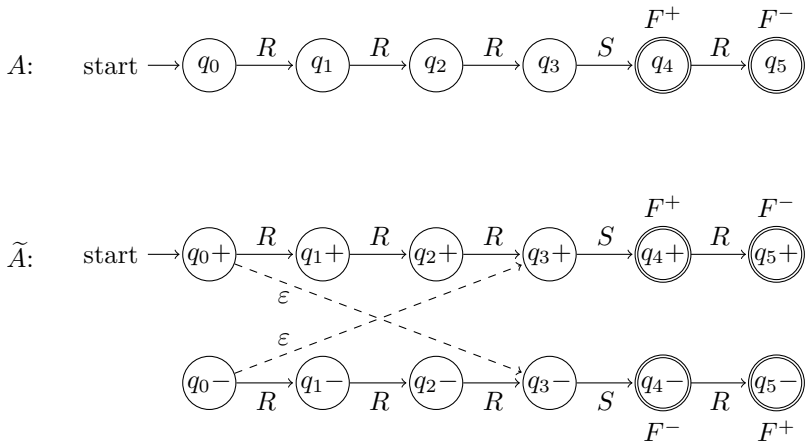

Figure 1 An example of an automaton A (above) and its corresponding automaton $\widetilde{A}$ (below). The final states from F^+ and F^- are marked by the labels F^+ and F^-, respectively.